

*ctf-writeups/picoCTF*



Web Exploitation  
Title: SSTI1  
Level: Easy



Author: levith4n

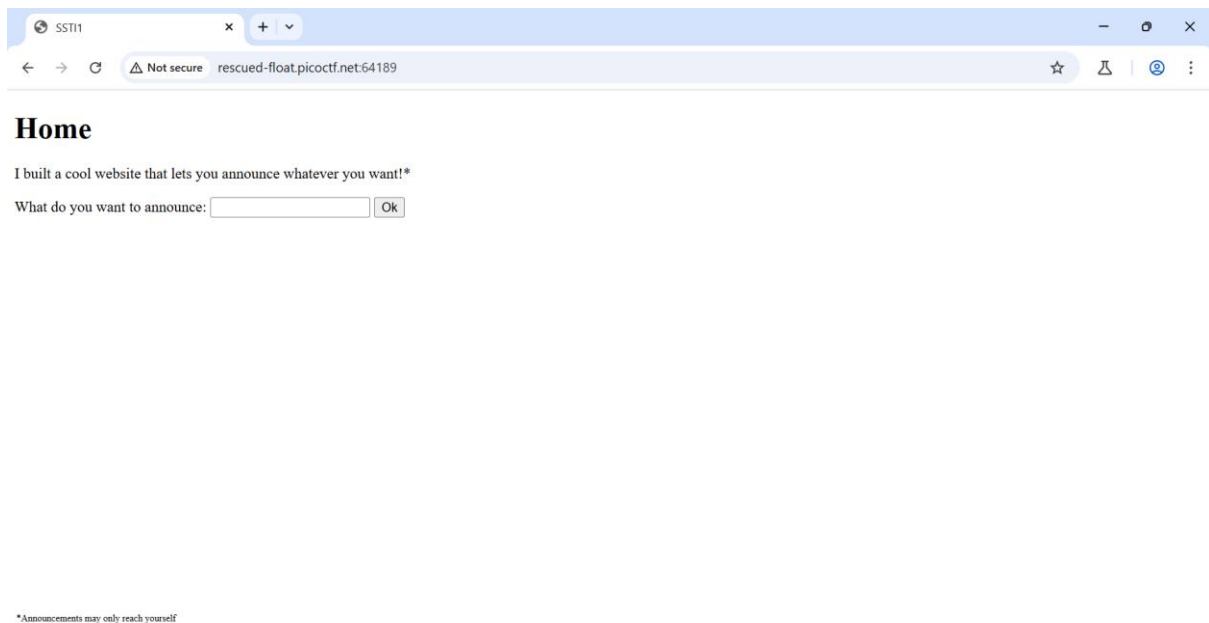
## Description

**Challenge Author: VENAX**

I made a cool website where you can announce whatever you want! Try it out! I heard templating is a cool and modular way to build web apps! Check out my website!

Hints 1: Server Side Template Injection

Ini adalah halaman utama dari website ini, berdasarkan instruksi yang diberikan, kita bisa menginputkan sebuah announce pada form yang disediakan.



SST11 x + v

← → ↻ ⚠ Not secure rescued-float.picocf.net:64189 ☆ 🏠 🌐 ⋮

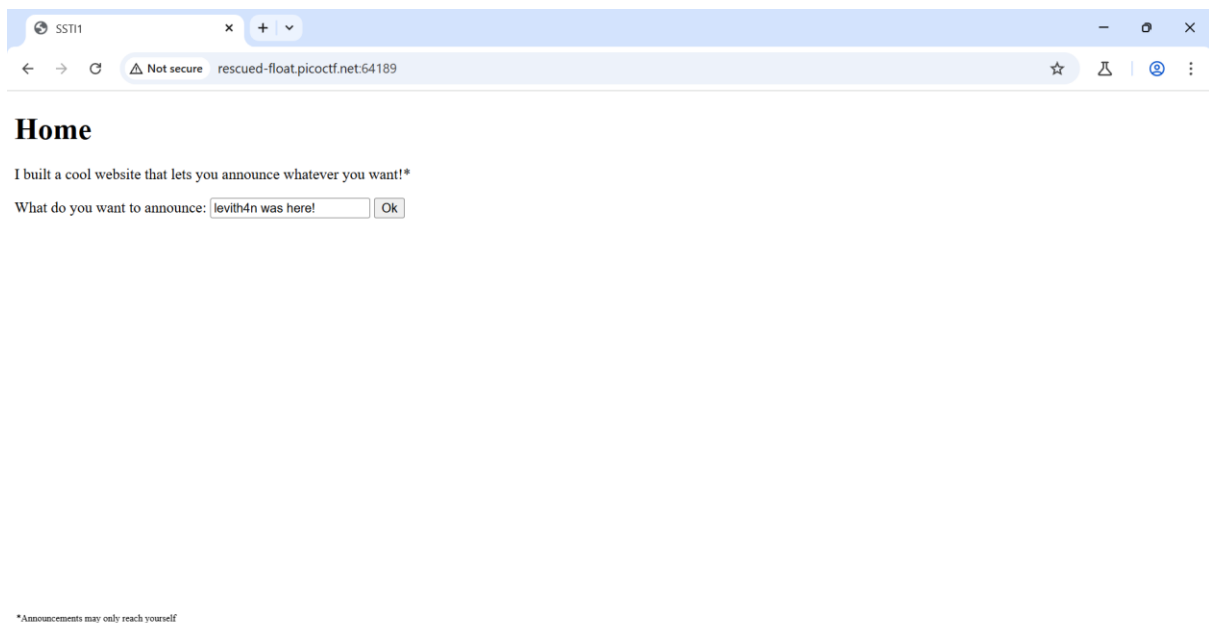
## Home

I built a cool website that lets you announce whatever you want!\*

What do you want to announce:

\*Announcements may only reach yourself

Kita dapat langsung mencoba memasukkan kalimat *“levith4n was here!”*.



SST11 x + v

← → ↻ ⚠ Not secure rescued-float.picocf.net:64189 ☆ 🏠 🌐 ⋮

## Home

I built a cool website that lets you announce whatever you want!\*

What do you want to announce:

\*Announcements may only reach yourself

Setelah mengirim input dimasukkan, aplikasi web meng-redirect ke path ‘/announce’ dan menampilkan announce yang kita inputkan sebelumnya.

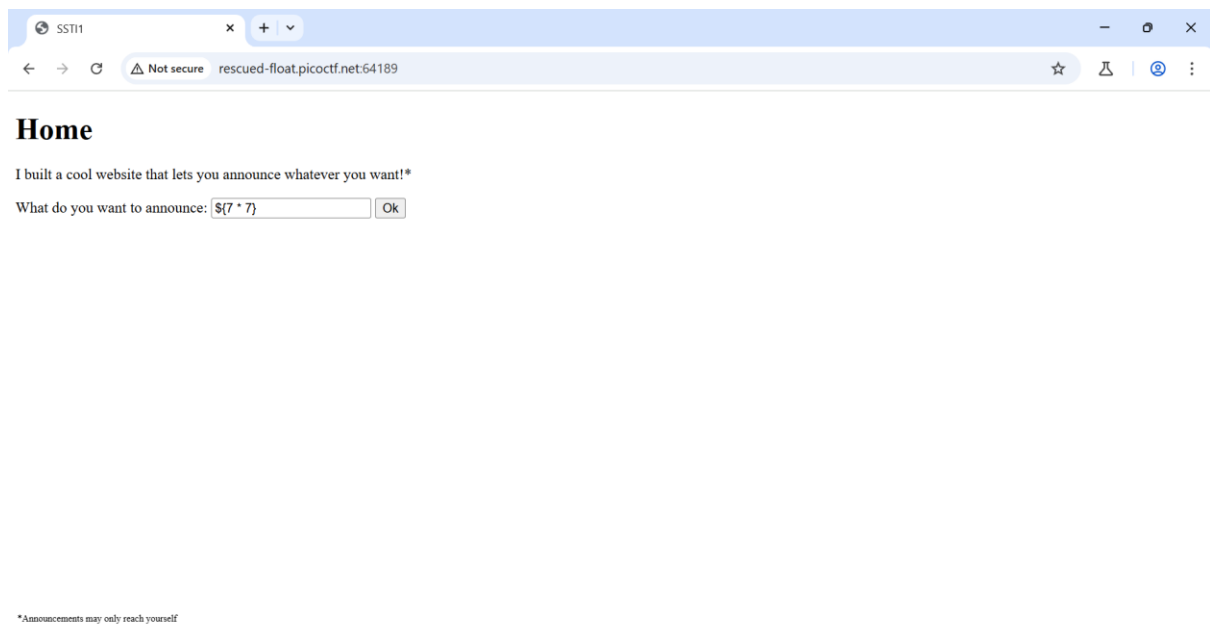


# levith4n was here!

Biasanya hal seperti diatas dibuat menggunakan template engine, tetapi untuk memastikan kebenarannya, kita dapat mengikuti diagram dibawah ini.



Kita coba masukkan payload pertama, yaitu  $\${7*7}$ .

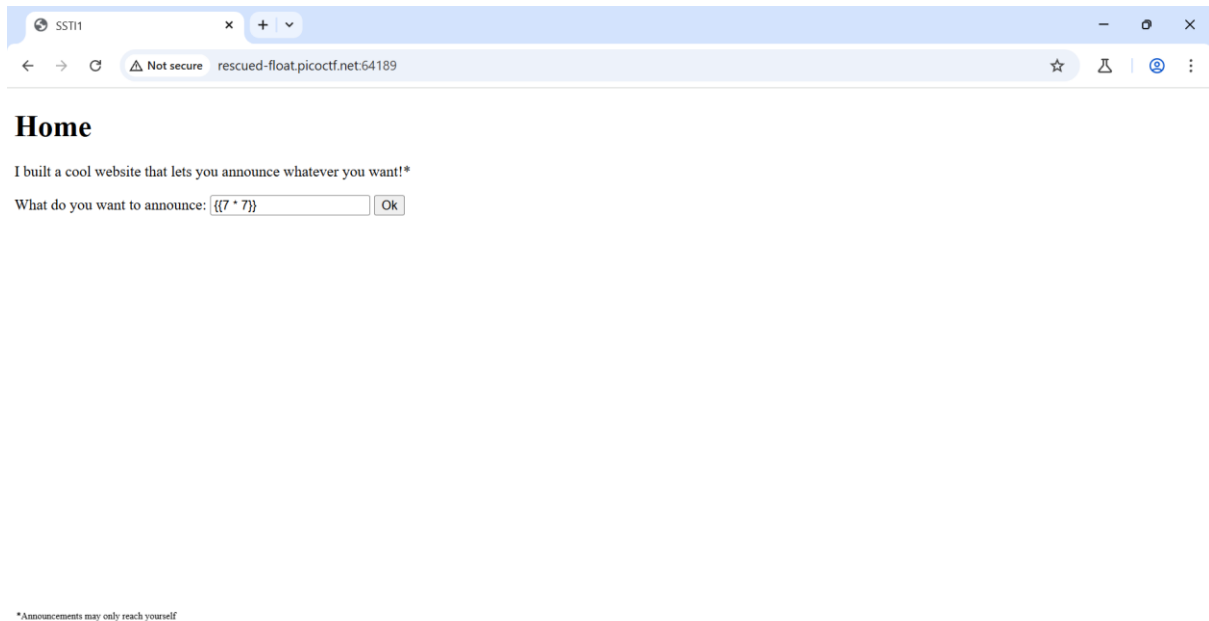


Ternyata nilai perkalian  $7*7$  tidak tereksekusi, sehingga kita akan mengikuti panah merah pada diagram.



$\${7 * 7}$

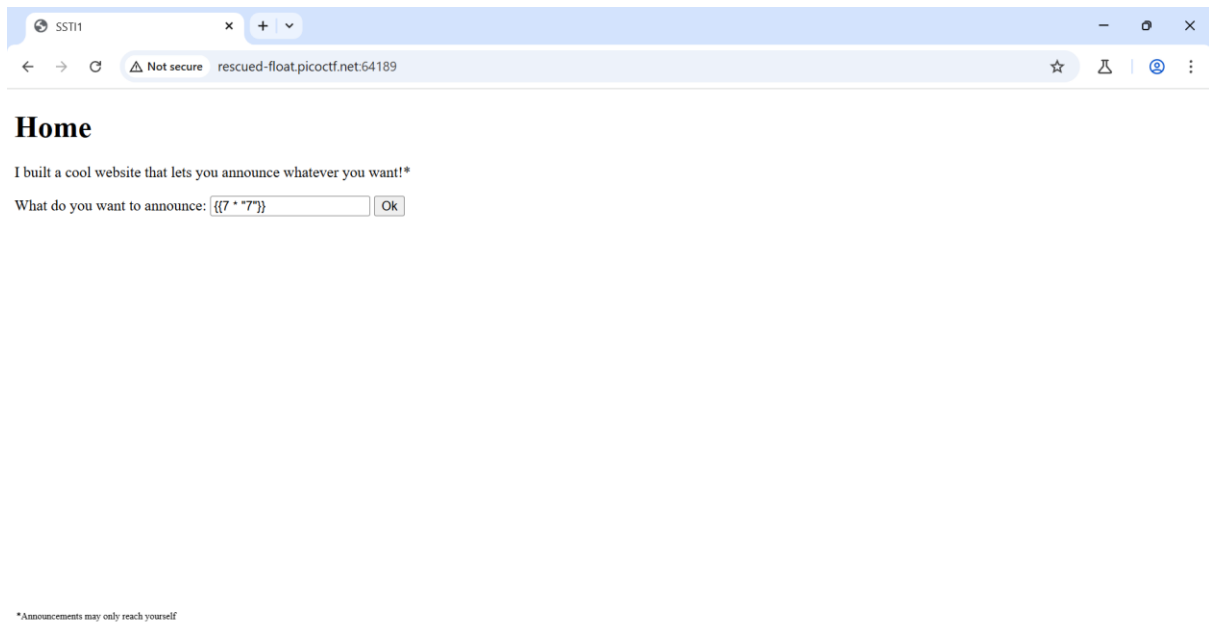
Selanjutnya, kita coba menggunakan payload kedua yang ditunjuk oleh panah merah, yaitu ‘ $\{\{7*7\}\}$ ’.



Ternyata hasil dari perkalian  $7 * 7$  tereksekusi, sehingga kita dapat mengikuti panah hijau selanjutnya.



Payload ketiga yang ditunjukkan oleh panah hijau adalah `{{7*'7'}}`.



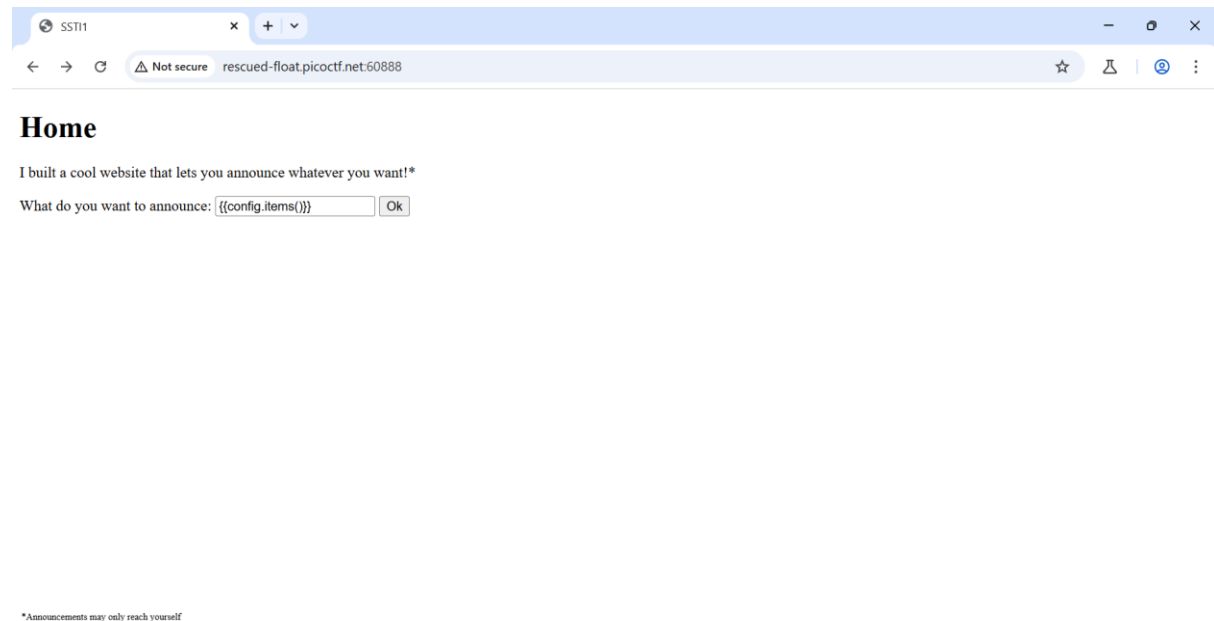
Ternyata hal ini berhasil tereksekusi, yaitu menampilkan strings 7 sebanyak tujuh kali. Berdasarkan diagram kita dapat mengasumsikan bahwa template engine yang digunakan adalah Jinja2 atau Twig.



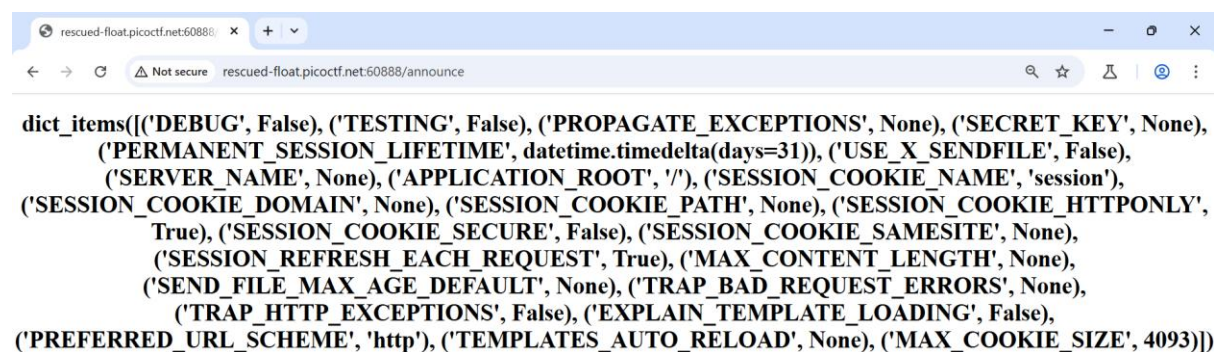
77777777

Untuk memastikan, kita dapat mencoba mengetes salah satunya terlebih dahulu, pada write up kali ini kita akan mengecek Jinja2 terlebih dahulu menggunakan payload `{{config.items()}}` guna melakukan Information Disclosure.

Payload tersebut bertujuan untuk menampilkan konfigurasi aplikasi web, 'config' adalah sebuah dictionary yang berisi konfigurasi aplikasi web, sedangkan 'items()' akan mengembalikan semua key-value pairs dalam bentuk daftar tuple (di Python 3, ini adalah objek 'dict\_items' yang bisa diiterasi).



Ternyata berhasil tereksekusi, ini menandakan bahwa aplikasi web menggunakan template engine Jinja2 dan melalui konfigurasi yang ditampilkan, **SECRET\_KEY** tidak diatur. Namun, fokus kita tidak pada hal ini, tetapi kita akan mencoba melakukan RCE.



Untuk melakukan RCE, kita perlu mengecek apakah Jinja2 menerapkan sandbox atau tidak (unsandbox) dengan payload ‘{{self.\_\_init\_\_.\_\_globals\_\_.\_\_builtins\_\_}}’. Jika output dari payload ini ditampilkan (list builtin function python), maka Jinja2 tidak menerapkan sandbox (unsandbox), jika output tidak ditampilkan maka sebaliknya (sandbox).



Ternyata berhasil tereksekusi dan menampilkan hasil yang diharapkan, hal ini menandakan bahwa Jinja2 tidak menerapkan sandbox (unsandbox). Sehingga kita dapat melakukan RCE dengan memanfaatkan builtin function ‘\_\_import\_\_’.



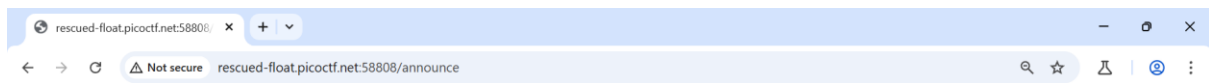


Selanjutnya, kita langsung mencoba melakukan RCE dengan payload `'{{self.__init__.__globals__.__builtins__.__import__('os').popen('ls').read()}}'`.

Payload ini memanfaatkan builtin function import untuk mengimpor modul os, sehingga kita dapat menggunakan function popen(). Function popen() dapat digunakan untuk berinteraksi dengan perintah sistem operasi, kemudian function ini mengembalikan file-like object, sehingga dapat diberlakukan method read().



Good, hal ini berhasil, kita menemukan file flag.



Terakhir, kita tinggal buka file flag sebelumnya menggunakan payload ‘{{self.\_\_init\_\_.\_\_globals\_\_.\_\_builtins\_\_.\_\_import\_\_(‘os’).popen(‘cat flag’).read()}}’.



Akhirnya, berhasil mendapatkan flagnya.



Thank you for reading!

